# SMART HOME DASHBOARD

## IOT PROJECT PRESENTATION

Mihai Serban

# AGENDA

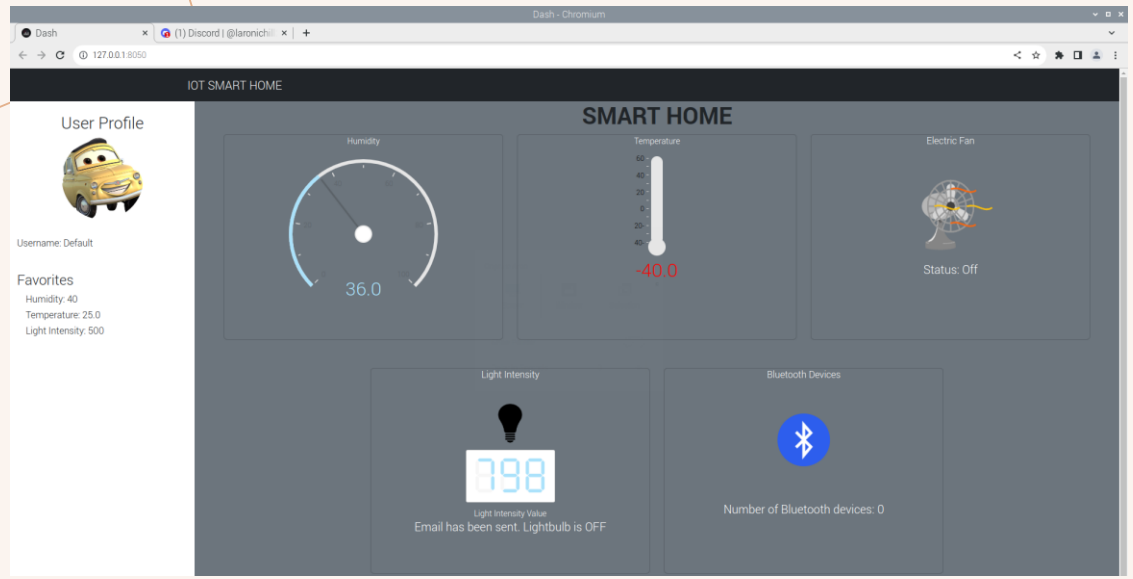
- **Project Overview**
- **Dashboard**
- **Project Wiring**
- **Individual Roles and Tasks**
- **Phases and their tasks**
- **Challenges and Accomplishments**

# PROJECT OVERVIEW

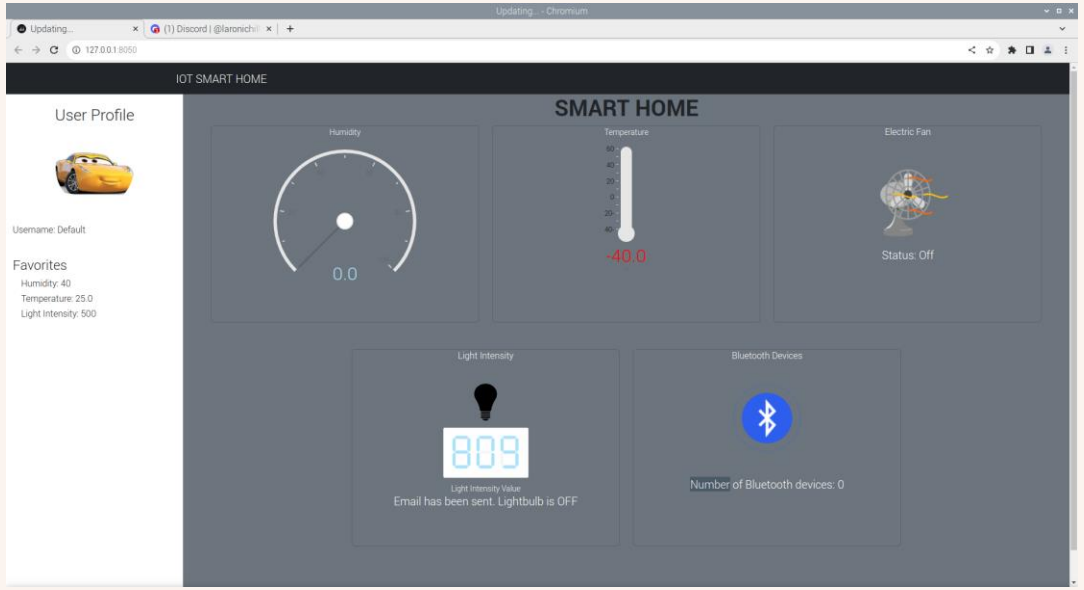
The IoT project focused on making a smart home. We collected data from sensors, sent it to the Raspberry Pi (Rpi), made decisions based on that data, and showed the results on the dashboard. The aim was to demonstrate how various parts of a smart home can work well together.

# DASHBOARD

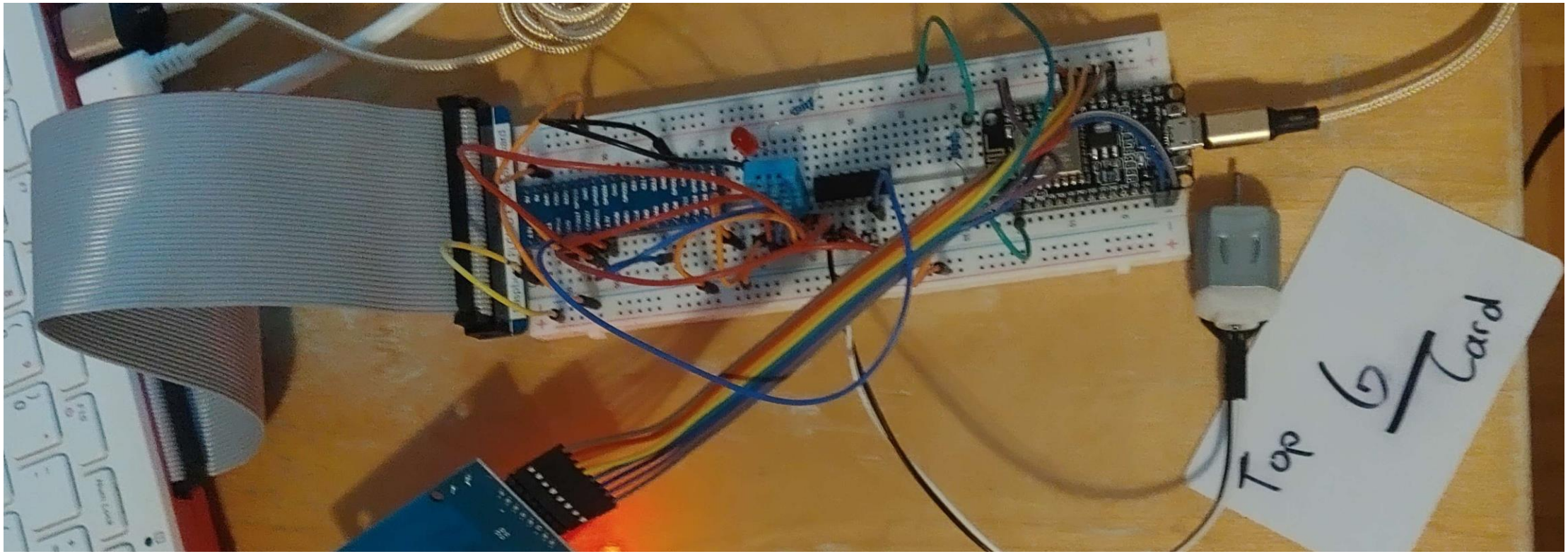
## Default




## RFID card tapped => User1



# THE PROJECT WIRING



# INDIVIDUAL ROLES AND TASKS

## Phase 1

Mihai

Initial Setup and basic dashboard

Jerico

Found Picture and click

Sayem

Overview and debug

## Phase 2

Mihai

Debugging and Wiring issues fixed

Jerico

Designed the wiring

Sayem

Added code to dashboard.py

# INDIVIDUAL ROLES AND TASKS

## Phase 3

Jerico and Sayem did most of the work

Mihai only helped debugging in class

## Phase 4

Mihai did everything

Documentation

Powerpoint

Dashboard and RFID addition

Github setup

Fritzing wiring

Etc.

# PHASES AND THEIR TASKS

## PHASE 1 - DESCRIPTION

In this phase, we create an IoT dashboard that can turn on and off a led on the breadboard from a simple click.

# PHASES AND THEIR TASKS

## PHASE 1 - UTILITIES

Components

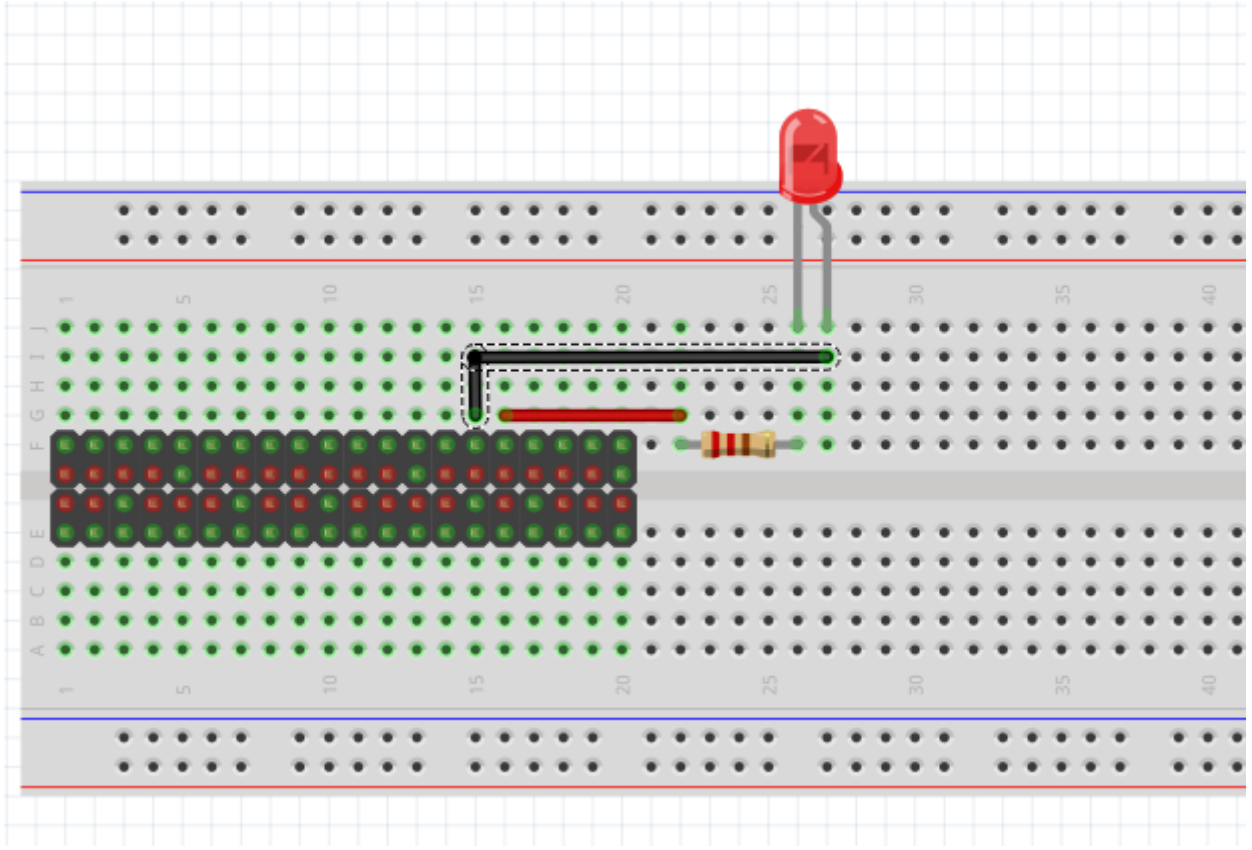
- Resistors
- Wires
- LED
- Breadboard
- Raspberry Pi

Libraries

- Dash
- dash_daq
- time
- RPi.GPIO

# PHASES AND THEIR TASKS

## PHASE 1 – WIRING



# PHASES AND THEIR TASKS

## PHASE 2 - DESCRIPTION

In this phase, we create an IoT dashboard displaying temperature and humidity using gauges. When the temperature exceeds a certain threshold, an automatic email is sent to the user, asking if they want to start the fan. If the user agrees, the system activates the fan and updates its status on the dashboard. If not, no action is taken.

# PHASES AND THEIR TASKS

## PHASE 2 - UTILITIES

Components

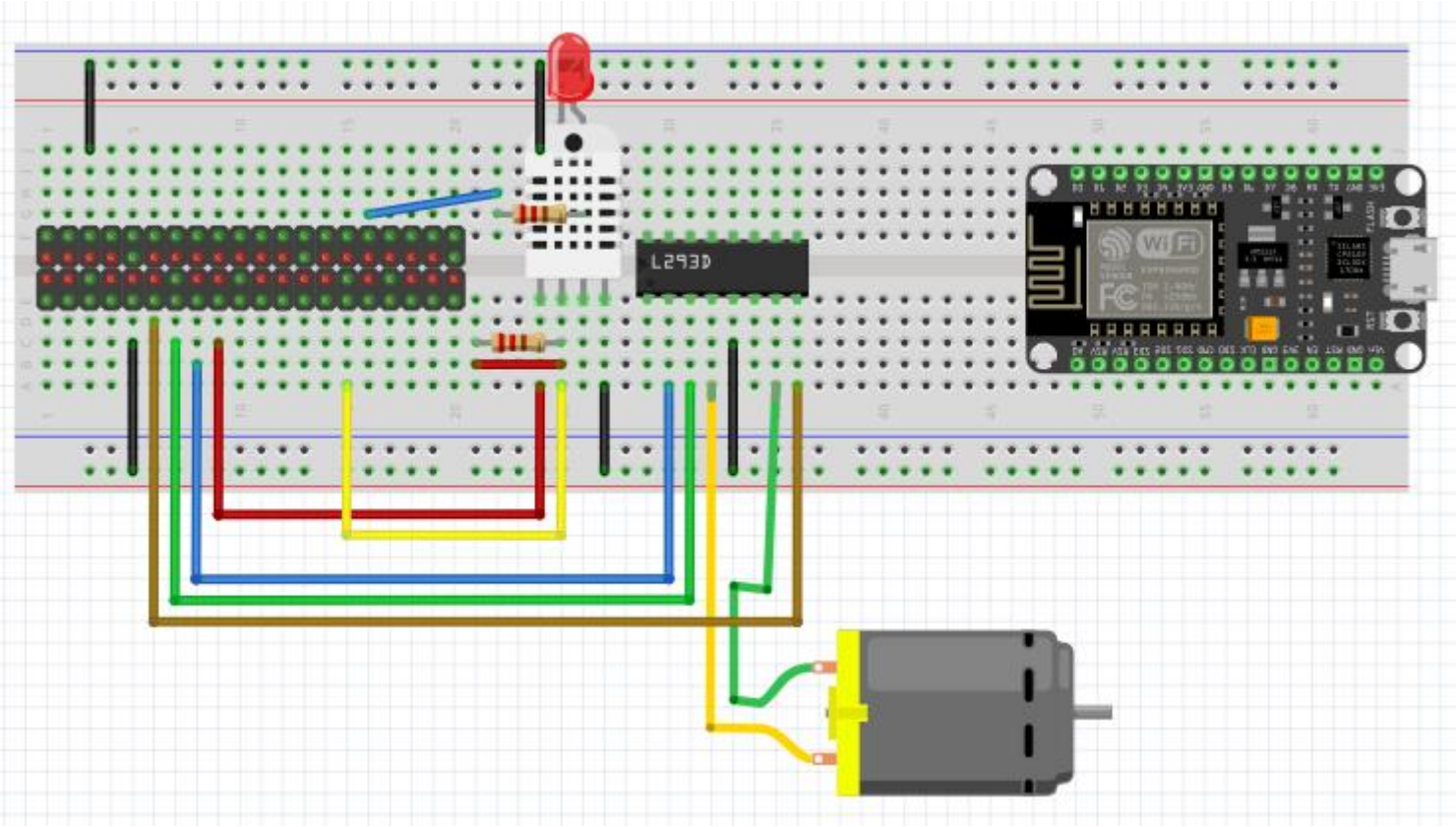
- DHT11 Temperature and Humidity sensor
- Resistors
- Wires
- DC motor
- IC L293D (Motor Driver)
- Breadboard
- Raspberry Pi

Libraries

- Dash
- datetime
- dash_daq
- time
- Freenove_DHT
- smtplib
- email
- Imaplib
- RPi.GPIO

# PHASES AND THEIR TASKS

## PHASE 2 – WIRING



# PHASES AND THEIR TASKS

## PHASE 3 – DESCRIPTION

In this part, we had to use a photoresistor with the ESP8266/ESP32 to measure light. We then sent this data to the Raspberry Pi and MQTT broker. If the light went below 400, we turned on an LED and sent an email. The dashboard had to show the current light level, its status, and confirm the email was sent.

# PHASES AND THEIR TASKS

## PHASE 3 - UTILITIES

Components

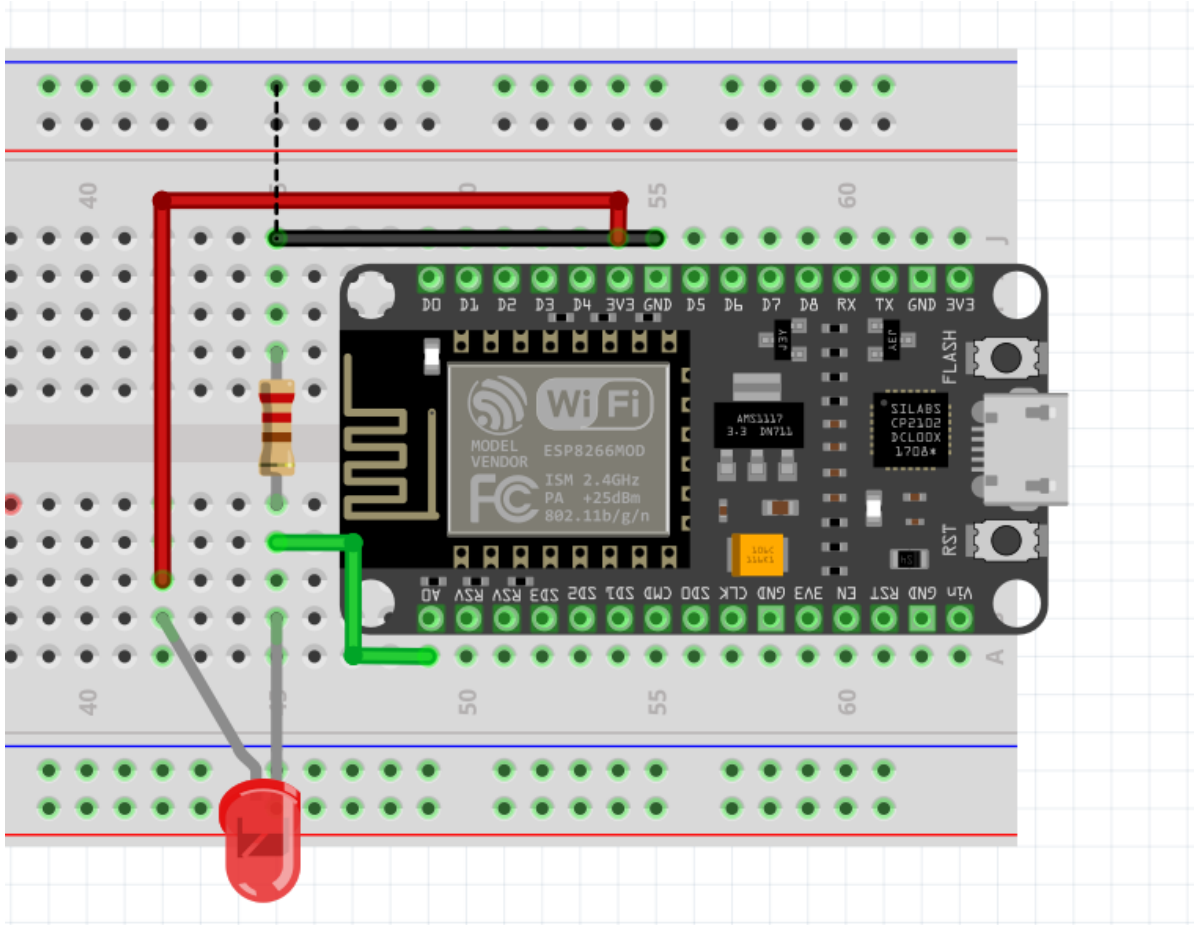
- Photoresistor
- Resistors
- Wires
- Breadboard
- Raspberry Pi
- ESP8266 or ESP32
- LED

Libraries

- Sqlite3
- paho.mqtt.client (mqtt)
- datetime
- time
- Freenove_DHT
- smtplib
- MIMEText
- MIMEMultipart
- email
- Imaplib
- ESP8266WiFi.h
- PubSubClient.h
- Arduino.h

# PHASES AND THEIR TASKS

## PHASE 3 – WIRING



# PHASES AND THEIR TASKS

## PHASE 4 - DESCRIPTION

- Task #1 involves creating user profiles with RFID tag numbers, temperature thresholds, and light intensity thresholds. When an RFID tag is read, the system updates thresholds based on the user's profile and sends an email notifying the entry time. The dashboard displays user profile information.

- Task #2 focuses on counting nearby Bluetooth-enabled devices.

# PHASES AND THEIR TASKS

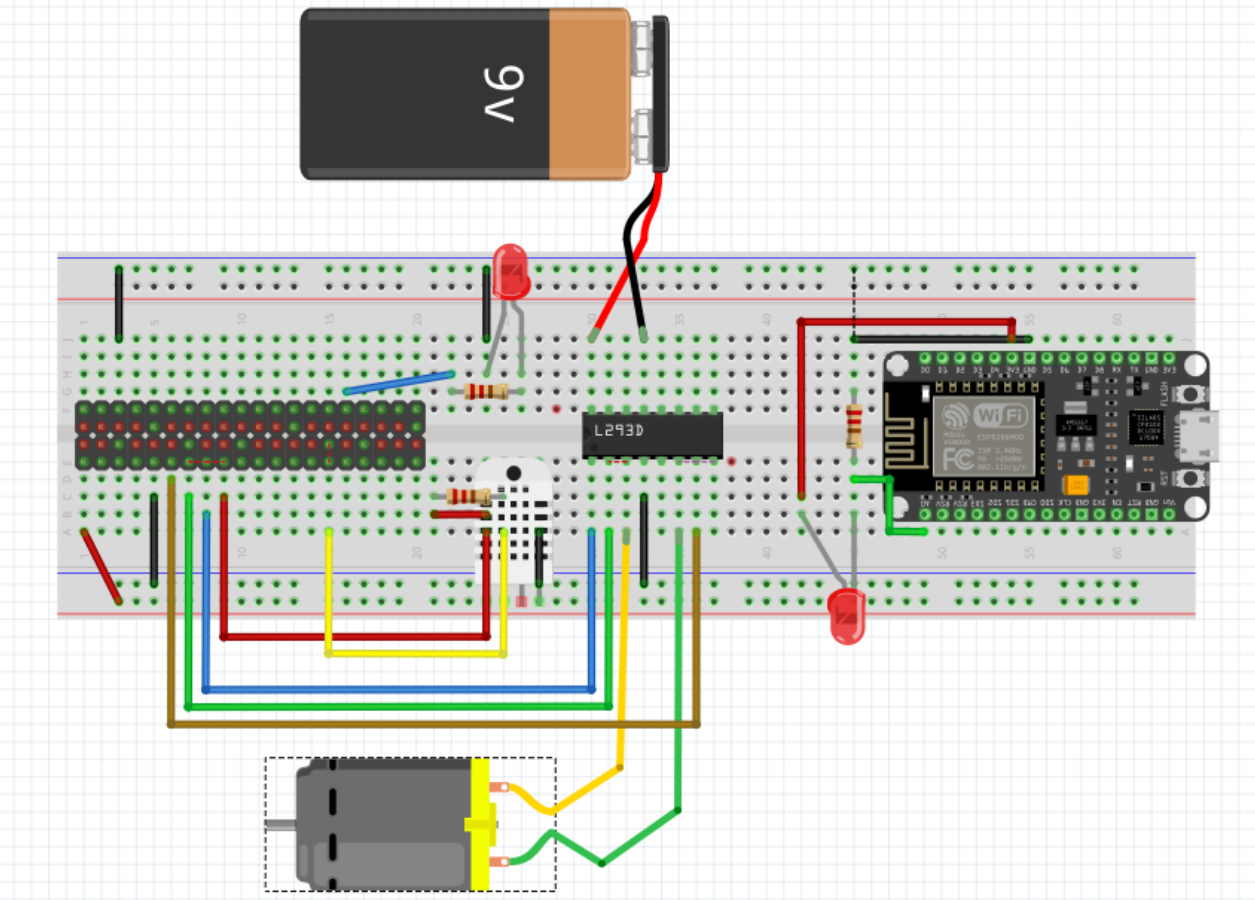
## PHASE 4 - UTILITIES

Components
- Photoresistor
- Resistors
- Wires
- Breadboard
- Raspberry Pi
- ESP8266 or ESP32
- LED
- RC522 RFID Module

Libraries
- Bluepy.btle
- Sqlite3
- datetime
- time
- Freenove_DHT
- smtplib
- MIMEText
- MIMEMultipart
- email
- Imaplib
- ESP8266WiFi.h
- PubSubClient.h
- Arduino.h

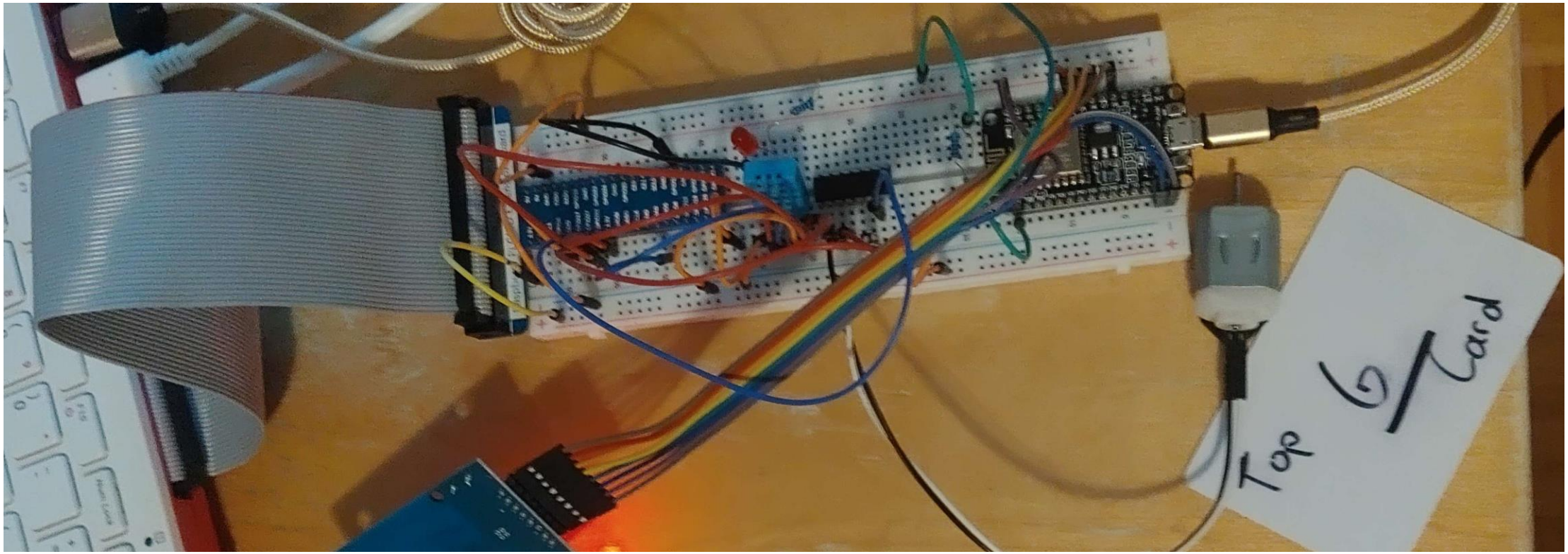
# PHASES AND THEIR TASKS

## PHASE 4 – WIRING



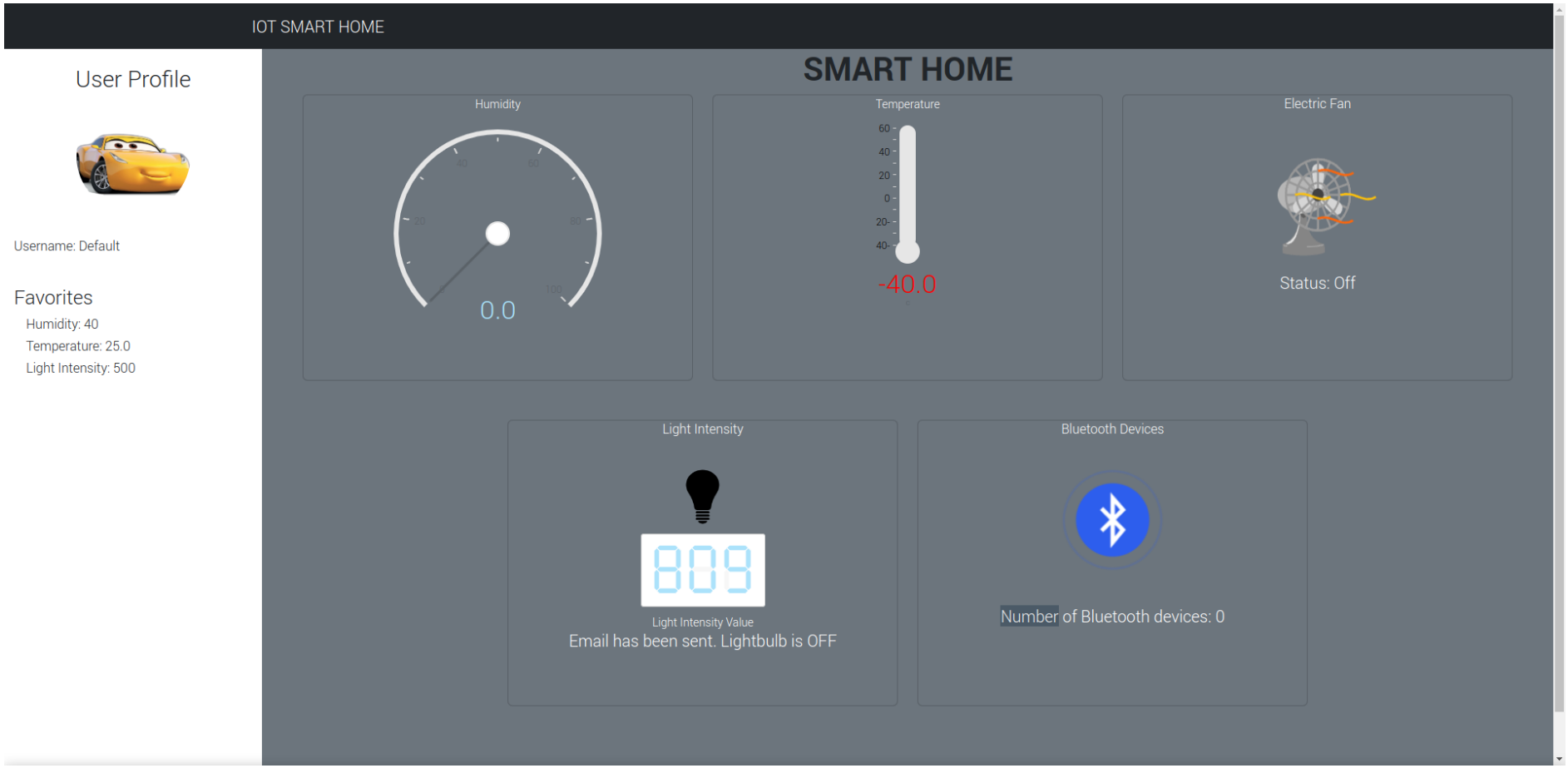
# PHASES AND THEIR TASKS

## PHASE 4 – IN REAL TIME



# PHASES AND THEIR TASKS

## PHASE 4 – DASHBOARD



# PROBLEMS ENCOUNTERED

After getting kicked, I had little to no data on our project apart from phase 2 and phase 3 code.

# CHALLENGES AND ACCOMPLISHMENTS

## PHASE 1

Installation of Dash was unclear when it came to the specifications and recommendations. Had to reinstall twice.

## PHASE 2

Writing code for the email reading was very difficult with Gmail as the documentation online was either off or obsolete.

## PHASE 3

Very difficult with no knowledge to set up the database

## PHASE 4

Incredibly slow dashboard. Impossible to debug unless you use a second computer. Microcontrollers burning out from having to rewire every time you work on it.

# CHALLENGES AND ACCOMPLISHMENTS

We've successfully engineered a sophisticated simulated smart home, integrating an assortment of advanced components—sensors, motors, a single-board computer, and microcontrollers. Our project unfolded in four distinct phases, encompassing the development of a smart thermometer system, an intelligent lighting setup, an occupancy detection system, and a personalized user profile module.

# THANK YOU

Mihai Serban

# WE MADE IT